Algorithm Complexity

Objective

The objective of this lab is to understand how we can analyze the complexity of an algorithm. Developed two algorithm for searching and find time complexity to analyze which one is better for a given data size.

Task

1. **Sequential Search**

This algorithm takes about N steps for an unsuccessful search and about N/2 steps, on the average, for a successful search.

1. **Binary search**

Divide the array into two parts, determine which of the two parts the key being sought belongs to, then concentrate on that part ... The total number of comparisons is only about lgN.

Problem 1

Write a program that empirically test the Olog(n) complexity of the binary search algorithm. For various array sizes n, generate a random array of integers in the range 0 to 2n. (Thus, about half the numbers in the range will be in the array, with possible duplication). For each of these arrays, sort the array, and then run the binary search for the target t, for each t from 0 to 2n. On each of these runs, count the number k of iterations of the algorithm’s while loop. For each n, print n, the average of the 2n values of k, and the value of logn. (The expression Math.log(n)/Math.log(2) returns the binary logarithm of n. Use System.currentTimeMillis() function to find actual time algorithm.). Your output should look like the following:

N Average log(n)

----------------------------------------------

4 2.125 2.0

8 3.063 3.0

.

.

.

Problem 2